NON-REPUDIATION IN WEB SERVICES INTERACTIONS

Elvis Ling Ing Seng, Vincent Khoo Kay Teong
Information Systems Engineering Research Group
School of Computer Sciences
Universiti Sains Malaysia, 11800 USM, Pulau Pinang, Malaysia.
Email: elviscss@yahoo.com, vkhoo@cs.usm.my

ABSTRACT

A Web service is a set of programmable functions that could be invoked and consumed through some Internet protocols. The design of Web services has been plagued with security flaws. Web services invocations could be executed on the fly with the invokers remain anonymous. The Web services creators tend to overlook the need to securely identify the identity of the Web services consumers and the confidentiality and integrity of the interacted entities and processes among the Web services. This paper attempts to model the potential threats and vulnerabilities that plague the interactions among Web services. The proposed model is presented as a set of UML sequence diagrams that show the critical points where Web services interactions could pose a severe threat to both the Web services consumers and creators. At the end of this paper, the need for identifying the entities and processes involved in a Web service interaction is highlighted through the recommendation of possible non-repudiation capabilities in Web services.

Keywords: Web service interaction, STRIDE, threats modelling, non-repudiation.

1.0 INTRODUCTION

The widespread use of the World Wide Web has created a common ground for many business organizations to trade among themselves regardless of their geographical locations. The World Wide Web has indeed crafted many successful stories for the business entrepreneurs who trade on the new borderless cyberspace. However, while the World Wide Web is making significant contributions to this progress, there remain many challenges to its further development into a safe environment where business trading can happen more securely. One area that needs an urgent attention is in Web services interactions.

For the Web Services to achieve its full potentials in improving business productivity, at least three key security requirements must be fulfilled namely, confidentiality, integrity, and availability [1] [2]. Confidentiality protects the identity of users involved in requesting and consuming Web services. However, confidentiality does not mean the anonymity of users as it is happening in the current Web services interactions. Integrity of Web services is only limited to the preservation of message authenticity, which indicates that the message returned by a Web service provider was not modified when it was being transmitted to the consumer. Availability of Web services will ensure its continuity in providing services to its service consumers.

This paper focuses on preserving the users and transactions confidentiality with some accounting mechanisms, and highlighting the issue of anonymous users in Web services interactions and how the capabilities of non-repudiation may resolve the problem.

2.0 WEB SERVICES INTERACTIONS

A Web Service can be easily located by making a request to a Web service registry [3], which acts as the centralized system to connect Web service consumers with Web service providers. Initially, a service provider will register its Web services at the Universal Description Discovery Integration (UDDI) registry. UDDI will process the Web services and publish the services into a searchable registry system.

When a request to discover and describe a Web service is made by a user, UDDI will locate the named service in its database. If the service is found, UDDI will post a list of available service providers who may satisfy the user request. At this point of the request, the return message to the user is wrapped in a Web Service Description Language (WSDL) to help the user to locate the required service provider [4] [5].

The user will continue the interaction workflow by making a Simple Object Access Protocol (SOAP) call to request the service provider to process the data contained in the message (also known as Web service invocation). The service provider is designed to response to this request [3] [4] upon receiving it and will proceed to process it and then return the result to the user, thus ending the Web service interaction. The interaction model is shown in Fig. 1.

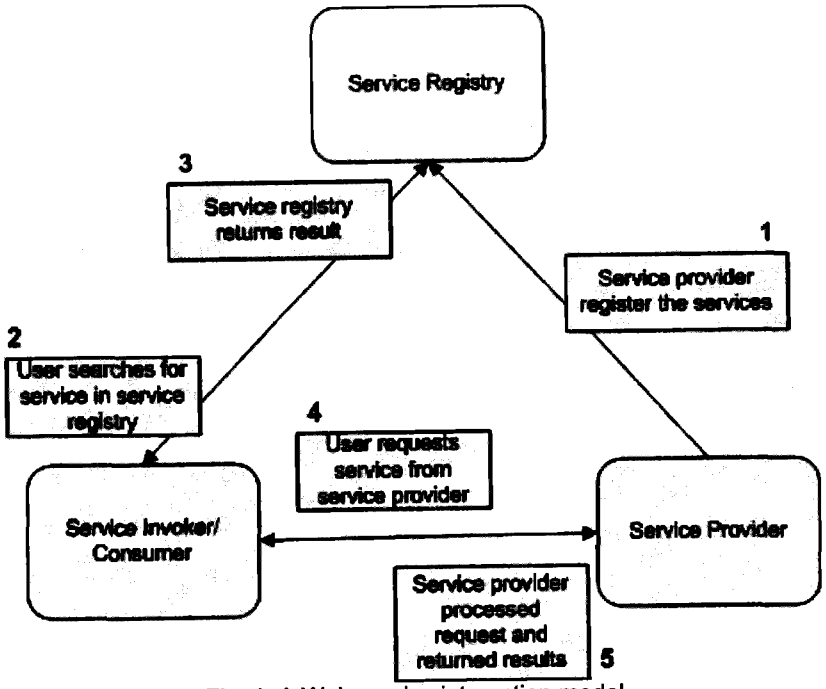

Fig. 1: A Web service interaction model

3.0 THREATS IN WEB SERVICES INTERACTIONS

As shown in the previous section, a Web service provider has no knowledge of the service consumer. Fig. 2 shows the Web service interaction model in Fig. 1 translated into a sequence diagram. From the diagram, one simple understanding that could be derived is that a Web service is invoked and consumed without the knowledge of the service provider. By the nature of its design, the service provider's main concern in a Web service interaction is to process the service request regardless of the request origin.

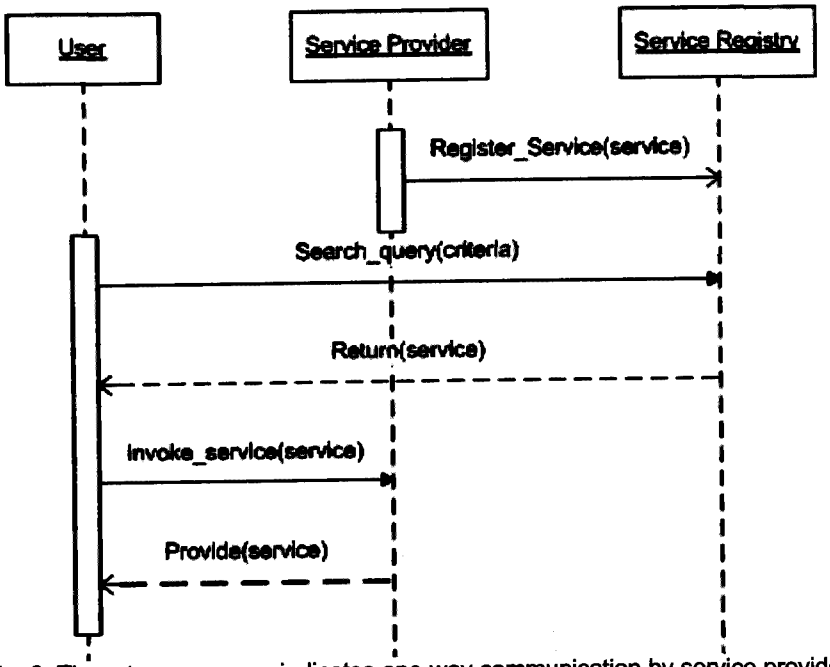

Fig. 2: The return message indicates one-way communication by service provider

As a result, the communication model between a service consumer and a service provider is a one-way communication, as indicated by the red dotted arrow in Fig. 2, where the Web service interaction ends at that point. The absence of a service acknowledgement from the users made it impossible for the service provider to identify the user identity.

According to the STRIDE threat model [6], six categories of threat can be identified in a Web service interaction namely: **S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service and **E**levation of privilege (hence the acronym STRIDE). These threats are identified at each point of a Web service interaction, and they involve all the entities in the interaction.

Referring to Fig. 3, the identities of service registry and service provider can be spoofed (number 1), where fraud services provided could lead to stolen data. Data integrity can be jeopardized while in transit (number 2), for example when a service provider's valid services are tampered while being sent to the service registry. Repudiation threat (number 3), which this paper attempts to mitigate, is identified when the user denied that the service provider has indeed processed the request and returned the results. In this case, the service provider is on the losing side as there is no mechanism to record the proof that the services have indeed been provided to the user.

As Web services travel the network in XML text, its content is exposed while being exchanged between the entities in a Web service interaction (number 4). The denial-of-service (number 5) and elevation of privilege (number 6) are the threats more commonly found at the Web servers that host the Web services, which could render services unavailability or permanent damages on the Web servers. Although it is agreeable that all these threats are present in a Web service interaction, this paper focuses only on the threat of repudiation.

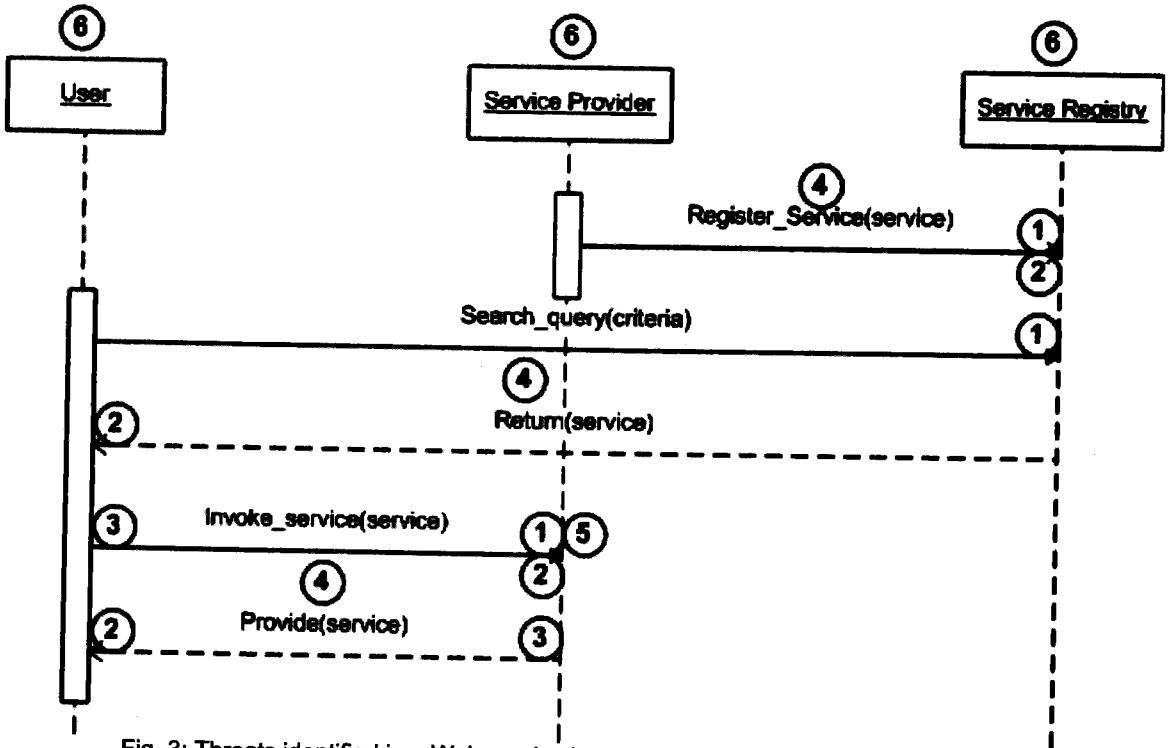

Fig. 3: Threats identified in a Web service interaction based on STRIDE threat model

Repudiation is the threat present in a Web service interaction because there is no mechanism for acknowledging that the service provider has provided its service to a service consumer [7]. The absence of a mechanism to record the Web services interactions has further exposed this threat. Although threat of repudiation can be easily mitigated with the use of XML Signature [8], the use of this feature in Web services has not been a mandatory requirement in its interactions.

As highlighted in [9], there are other threats and challenges that are bound in the context of Web service security, which include:

a. Message security between Web service intermediaries
b. Unauthorized access
c. Parameter injection
d. Message eavesdropping and replay
e. Firewalls evasion
f. Platform immaturity

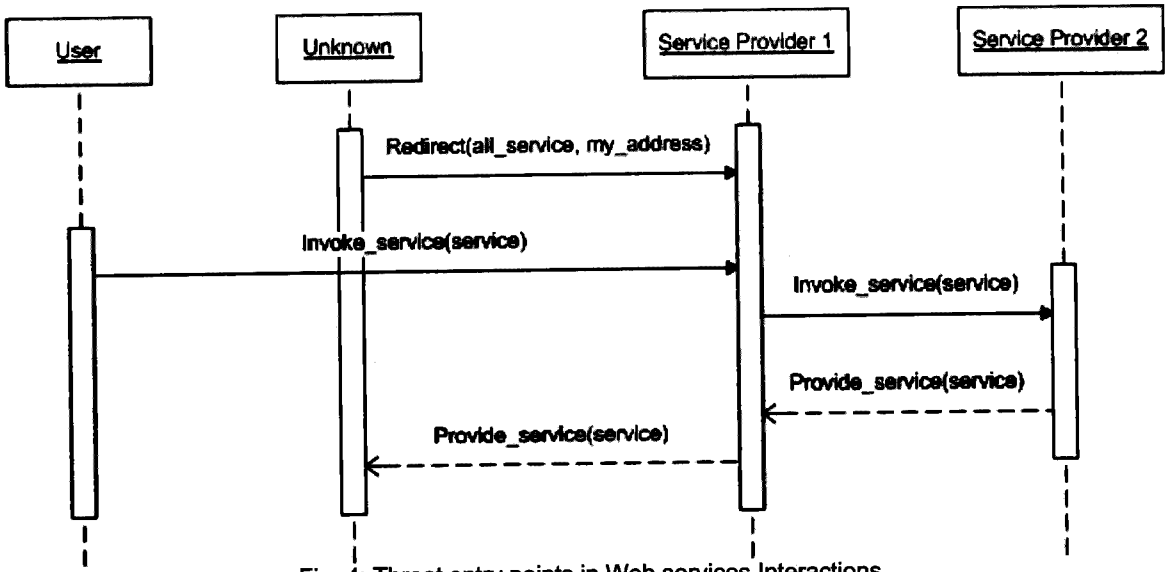

Fig. 4: Threat entry points in Web services Interactions

The threat analysis on a Web service interaction shows several threat entry points as shown in Fig. 4. Firstly, the attackers involved in a Web service interaction will perform Domain Name System (DNS) cache poisoning on Service Provider 1 so that all Web services deliveries will be redirected to the attacker. This attack is based on the vulnerability exists in a DNS system that may allow the attacker to send invalid packets of data to be cached into the DNS server that belongs Service Provider 1.

When a user makes a service request to a valid service provider, the request may be forwarded to another service provider for further processing. A spoofing threat point can be identified here where the request made by the user may be forwarded to a fake service provider, which can be an attacker itself. The attacker could have crafted a return message that may intently damage the systems located at the user end. This is identified as a tampering threat.

Back to the previous scenario, when Service Provider 1 delivers a service back to the user, the delivery path is redirected to the attacker, thus creating a repudiation attack whereby Service Provider 1 claims the service delivery when it actually never reaches the user. The interactions with threat of repudiation as depicted in Fig. 4 shows that the attack can be alleviated with a mandatory proof of delivery and proof of receipt.

4.0 NON-REPUDIATION

Non-repudiation is the term used in describing the nature of a system where participation and non-participation in an electronic event is deemed undeniable [10]. Two main capabilities of non-repudiation are to provide proof of delivery [11] and proof of receipt [12]. Therefore the goal of non-repudiation is to provide a fair exchange in electronic transactions whereby the source that initiated a transaction can be protected from a fraudulent allege made by its destination and vice versa [13] [14].

4.1 Non Repudiation in Web Services

In practice, every computer that communicates via a network system should be accounted for the course of actions taken along the network path. Since the wild nature of World Wide Web attracts malicious computer users to bring chaos into the cyberspace, it is therefore acceptable that the presence of some accounting mechanisms in Web Services should reduce the risk of consuming Web Services anonymously.

The use of an accounting system in the Web services interactions could prevent the consumption of Web services anonymously as shown in Fig. 5. The differences of this interaction are the service acknowledgement made by the service consumer and the recording of user request at the initial stage of service invocation.

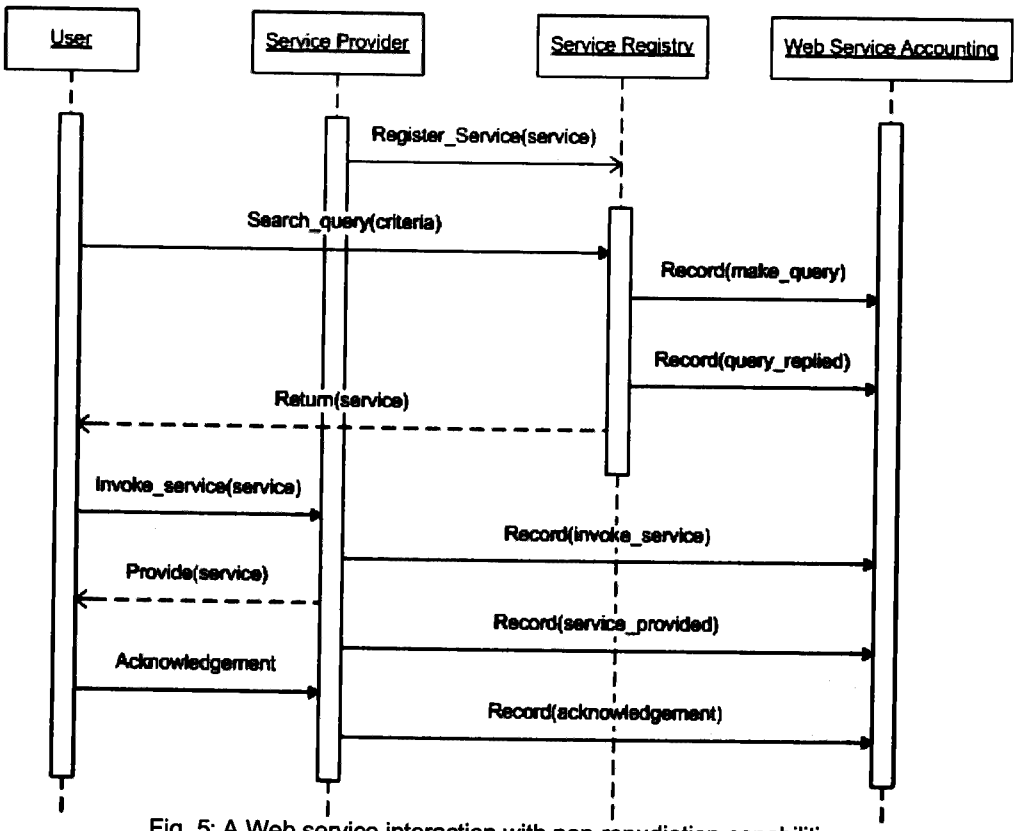

Fig. 5: A Web service interaction with non-repudiation capabilities

4.2 Accounting Mechanisms and Service Acknowledgements

The accounting mechanism in Web services interactions should comprise of a log archival [15] and other non-repudiation capabilities which include receipt acknowledgement and delivery notification. As shown in Fig. 5, whenever the users make up a service query to the Service Registry, the query is recorded into a Web service accounting system repository. If the query produces results, the Service Registry will return a list of service providers, while recording this into the accounting system repository.

Then, the user proceeds with the interaction by invoking a service from the service provider. As the user is the initiator of this interaction, every action taken shall be recorded as part of the repudiation threat mitigation, providing an evidence base for future references for both the user and the service provider. When the service provider has processed the data as contained in the user's request, the action is recorded in the accounting system as a proof of delivery.

However, the interaction flow does not end there as shown in Fig. 5, where the service provider waits for a service acknowledgement from the user once the data has been completely processed. The acknowledgement sent by the user is in fact a proof of receipt that will be kept by the service provider. The accounting system will also have a copy of this acknowledgement in corresponding to the proof of delivery that was previously recorded when the service provider delivered the service to the user.

5.0 CONCLUSION AND FUTURE WORK

Identifying threats is crucial in developing secure Web services interactions. With the growing number of organizations leveraging the capabilities of Web services, it has been noted that the use of transaction logs is not enough in securing the online business transactions. More robust accounting mechanisms should be implemented in securing Web services interactions. Moreover, the capabilities of non-repudiation should be incorporated into every system on the Web.

More intensive research work needs to be done to establish the requirements of a new version of accounting mechanism that surpasses the outdated audit log that has become a mainstream for any system that needs to keep track of electronic transaction interchange. One immediate research problem is how the entities involved in Web services interactions record their activities and how these records could be secured and managed. Therefore, it is hoped that the design and implementation of a robust and trustworthy accounting mechanism could thwart the attempts to repudiate the services provided by service providers.

REFERENCES

[1] B., Roberta, CISSP Security Management and Practices, QUE Publishing, December 2002.

[2] R., J., Boncella, Web Services and Web Service Security, Proceedings of the Americas Conference in Information Systems, NY, August 2004.

[3] B., David, H., Hugo, M. C., Francis, N., Eric, C., Michael, F., Chris, and O., David, Web Service Architecture, W3C Working Group Note, February 2004.

[4] S., Vinoski, Web Services Interaction Models, Part 1: Current Practice, IEEE Internet Computing 6(3), May 2002, pp. 89-91.

[5] S., Vinoski, Putting the "Web" into Web Services: Interaction Models, Part 2, IEEE Internet Computing 6(4), May 2002, pp. 90-92.

[6] L., Desmet, B., Jacobs, F., Piessens, and W., Joossen, Threat Modelling for Web Services Based Web Applications, Eighth IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS 2004), September 2004, UK.

[7] J., Onieva, J., Zhou and J., Lopez, Non-repudiation protocols for multiple entities, Computer Communications, 27(16):pp. 1608-1616, October 2004.

[8] J., Rosenberg and D., Remy, Securing Web Services with WS-Security – Demystifying WS-Security, WS-Policy, SAML, XML Signature and XML Encryption, SAMS Publishing, Indianapolis, 2004.

[9] J., Holgersson and E., Soderstrom, Web Service Security – Vulnerabilities and Threats within the Context of WS-Security, The 4th Conference on Standardization and Innovation in Information Technology, pp. 138-146, September 2005.

[10] J., Zhou and D., Gollmann, Evidence and Non-repudiation, Journal of Network and Computer Applications, Academic Press Ltd., 1997, pp. 267-281.

[11] M., Wichert, D., Ingham, and S., Caughey. Non-repudiation Evidence Generation for CORBA using XML, Proc. IEEE Annual Comp. Security Applications Conf., Phoenix, USA, 1999.

[12] T., Coffey, P., Saidha, Non-repudiation with mandatory proof of receipt, Computer Communication Review, 26(1), pp6-18. 1996.

[13] J., Zhou and D., Gollman, A fair non-repudiation protocol, Proc. 1996 Symp. on Research in Security and Privacy, pp. 55-61, Oakland, CA, USA, 1996.

[14] J., Cederquist, R., Corin and M., Torabi Dashti, On the Quest for Impartiality: Design and Analysis of a Fair Non-repudiation Protocol, Lecture Notes in Computer Science 3783, Proceedings of 2005 International Conference on Information and Communications Security, pp. 27-39, Beijing, China, December 2005.

[15] M., H., Shao, A Study on Chain of Evidence Supporting Disputes Resolution in Electronic Commerce, PhD Thesis, National Chiao Tung University, Taiwan, 2005.

BIOGRAPHY

Elvis Ling Ing Seng graduated with a Bachelor's Degree in Computer Science from Universiti Sains Malaysia in May 2006. Currently, he is registered as a part-time research student in the Master of Computer Science programme while working in Intel. His research interests include security mechanisms and countermeasures for Web-based applications.